

AD-A102 160

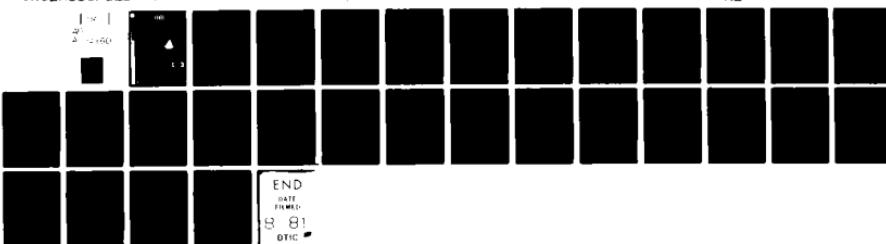
CALIFORNIA UNIV BERKELEY OPERATIONS RESEARCH CENTER  
MULTIPLE RESOURCE SMOOTHING IN SHIPYARD PLANNING. (U)  
JUN 81 R C LEACHMAN, J BOYSEN  
ORC-81-15

F/G 12/2

UNCLASSIFIED

N00014-76-C-0134

NL



END  
DATE  
FILED  
8 81  
OTIC

ADA102160

# LEVEL II

ORC 81-15  
JUNE 1981

12

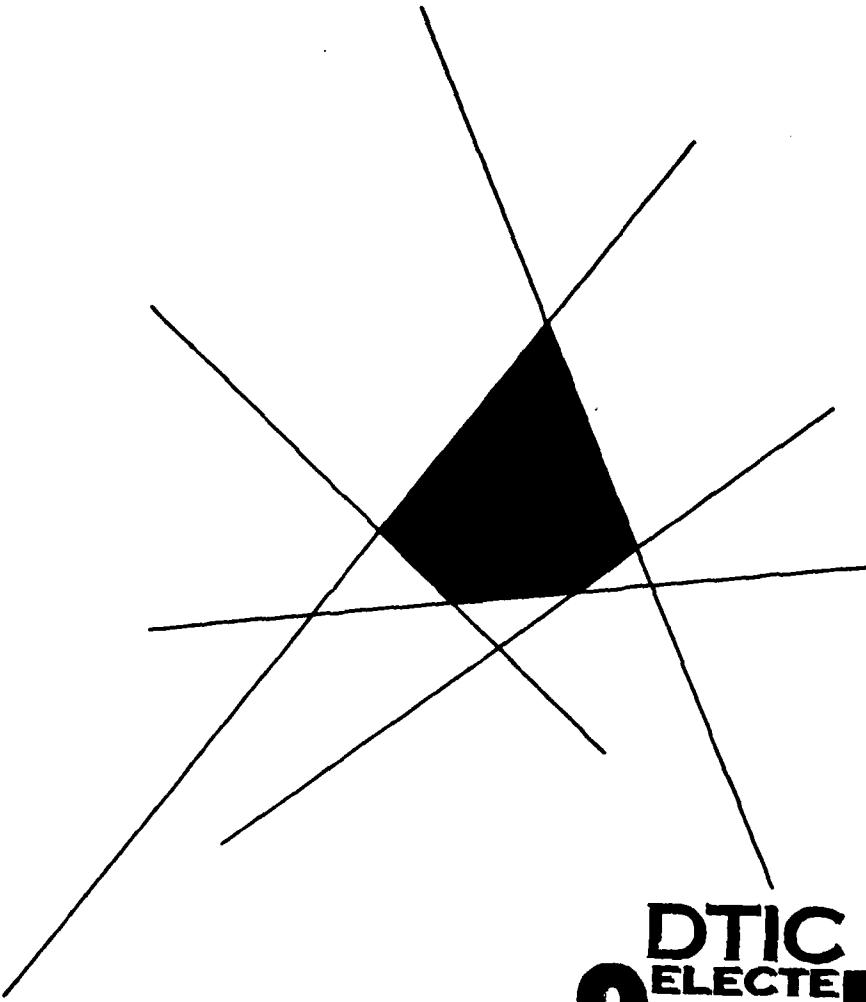
MULTIPLE RESOURCE SMOOTHING IN SHIPYARD PLANNING

by

ROBERT C. LEACHMAN

and

JOERG BOYSEN



DTIC  
SELECTED  
S JUL 29 1981 D  
F

OPERATIONS  
RESEARCH  
CENTER

AMERICAN  
FILE COPY

UNIVERSITY OF CALIFORNIA · BERKELEY

817 29 038

## MULTIPLE RESOURCE SMOOTHING IN SHIPYARD PLANNING

by

Robert C. Leachman  
Department of Industrial Engineering  
and Operations Research  
University of California, Berkeley

and

Joerg Boysen  
Operations Research Center  
University of California, Berkeley

Accession For  
MILITARY  
TECH 748  
Unannounced  
Specification

JUNE 1981

OBC 81-15

This research was supported by the Office of Naval Research under Contract N00014-76-C-0134 with the University of California. Reproduction in whole or in part is permitted for any purpose of the United States Government.

Unclassified

**SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)**

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER ORC-81-15	2. GOVT ACCESSION NO. AD-A102 160	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) MULTIPLE RESOURCE SMOOTHING IN SHIPYARD PLANNING	5. TYPE OF REPORT & PERIOD COVERED Research Report.	
7. AUTHOR(s) Robert C. Leachman and Joerg Boysen	6. PERFORMING ORG. REPORT NUMBER N00014-76-C-0134	
9. PERFORMING ORGANIZATION NAME AND ADDRESS Operations Research Center University of California Berkeley, California 94720	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS NR 047 033	
11. CONTROLLING OFFICE NAME AND ADDRESS Office of Naval Research Department of the Navy Arlington, Virginia 22217	12. REPORT DATE June 1981	
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	13. NUMBER OF PAGES 29	
16. DISTRIBUTION STATEMENT (of this Report)	15. SECURITY CLASS. (of this report) Unclassified	
	15a. DECLASSIFICATION/ DOWNGRADING SCHEDULE	
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Resource Leveling Critical Path Analysis Capacity Planning Nonlinear Programming Construction Systems	Manpower Allocation	
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) (SEE ABSTRACT)		

DD FORM 1 JAN 73 1473

EDITION OF 1 NOV 68 IS OBSOLETE  
S/N 0102-LF-014-6601

Unclassified   
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

#### ABSTRACT

A heuristic approach for planning multiple resource workloads in project networks is discussed, with particular attention to ship overhaul planning. Projects are modeled using a critical-path-analysis activity network. Activity resource requirements are taken as given and activity loading is assumed to be constant. Activity duration variables are defined which then determine the demand rates per resource per activity.

An iterative nonlinear programming procedure assigns activity durations (and consequently, activity resource demand rates) to minimize resource peaking subject to meeting project due dates. Computational experience is described for application to a major ship overhaul.

## Table of Contents

	Page
I. Introduction and Scope of Problem.....	1
II. Construction Project Model .....	5
III. Resource Leveling.....	6
A. Peak Pricing Procedure .....	6
B. Optimization Problem Formulations.....	8
C. Procedure Termination and Cycling.....	11
D. Capacity Requirements vs. Project Delay .....	13
IV. Applications and Computational Experience.....	14
A. Network and Data.....	14
B. Experiments.....	14
V. Summary and Analysis of Results.....	22
Glossary of Notation .....	23
References.....	24

## **List of Exhibits**

	Page
1. Multi-Overhaul Scheduling Chart.....	2
2. Example of Key-op Activity .....	3
3. Resource Peak Intervals .....	7
4. (a) Single-Element Slack Paths .....	10
(b) Redundant Slack Paths.....	10
5. Optimization Problem Constraint Sets .....	12
6. List of Labor Shops .....	15
7. Example of Aggregate Activity.....	16
8. Example of Shop Interaction.....	17
9. Experimental Results: Shop Capacity Requirements.....	19
10. Experiment (b): Sequence of Load Histories. Shop 5.....	20
11. Comparison of Shop 8 Load Histories, Experiments (b), (c), (d).....	21

## I. Introduction and Scope of the Problem

Production activity in a naval shipyard consists of a variety of large ship overhaul projects. Each of these projects requires labor input in the tens or even hundreds of thousands of man-days from among 20 different trade shops. Project durations range from 3 months to 2½ years. Manpower loading on a ship overhaul is typically very peaked, and there is competition among various projects for manpower, dry docks, and other key resources. The planning and scheduling of projects so as to meet target completion dates while maintaining high levels of yard productivity is a challenging management problem.

In current practice, the problem is decomposed into two stages. First, an aggregate scheduling effort is undertaken, in which the "rough" scheduling of projects is made. In this effort, starting, ending, and a few intermediate *milestone* dates are established for each project. The goal of aggregate scheduling is to determine milestone dates which are timely but feasible, considering projections of available production manpower and dry docks.

Detailed information regarding resource and duration requirements for each project would be desirable to accomplish aggregate scheduling. But only very aggregate, preliminary, information is currently available to planners. A time history of the total man-days per working day is estimated from past experience for each project; by setting start dates for each project, these histories may be summed to estimate the total man-day history on the yard. By trial and error, milestone dates are developed for which the total load history is comparable to the projected available work force, and dry dock conflicts are avoided. (See Exhibit 1.) This load information is not broken down by labor shop, so that shop load infeasibilities may still arise.

In the second stage, detailed project scheduling of small component production activities known as *key-ops* is undertaken. The specification of these activities, numbering in the thousands for each overhaul, may require a year or more. A critical path network is then developed consisting of the key-op activities, which are then scheduled using critical path methods (CPM). In the scheduling effort network slack is uniformly allocated among activities, which tends to reduce the project's peak resource loads. See [3] for a discussion of CPM and resource load leveling methods. See Exhibit 2 for an example of a key-op description.

Milestone dates and key-op schedules are distributed to line (shop) production management, which is responsible for the day-to-day allocation of working crews and job supervision. In practice, adherence to key-op schedules is rare, but observance of milestone dates is accomplished where feasible. In a sense, line management accomplishes its own version of resource leveling by varying crew sizes and job assignments.

It is our opinion that aggregate scheduling techniques utilizing *more* detail than now used, coupled with scheduling techniques for individual projects based on *less* detailed activity

EXHIBIT 1  
MULTI-OVERHAUL SCHEDULING CHART

PROJECTS	MONTHLY PROJECT LOADS				
	FEB	MAR	APR	MAY	JUN
PROJECT A	2031	1648	1289	1048	854
PROJECT B	2059	1896	1182	856	539
PROJECT C	1488	1759	2126	1652	1467
PROJECT D			234	1175	2077
OTHER	410	390	380	370	370
TOTAL LOAD	5988	5693	5211	5101	5307
WORKFORCE	4975	4906	4698	4743	4486
OVERLOAD/ (UNDERLOAD)	1013	787	513	358	821

ALL FIGURES ARE MAN-DAY TOTALS,  
IRRESPECTIVE OF LABOR TYPE.

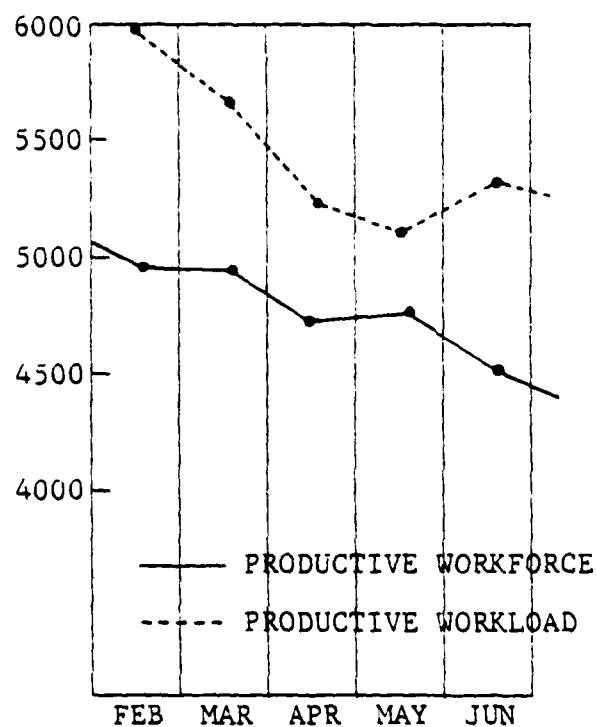


EXHIBIT 2  
EXAMPLE OF KEY-OP ACTIVITY

ACTIVITY 16662-26301-603      INSTALL SHAFT LUBE PUMPS

DURATION: 15 DAYS

SHOP HOURS REQUIRED

SHOP 5 (SHIPBOARD MECHANICAL)      64 MAN HOURS

SHOP 7 (ELECTRICAL CABLES)      8 MAN HOURS

SHOP 8 (PIPEFITTERS)      48 MAN HOURS

4

definition than now pursued, would be valuable. Shop capacities should be explicitly treated in aggregate scheduling. Work scheduling should not be done at the key-op level of detail, since such fine scheduling is largely ignored in actual operations. An Operations Research model used in shipyard project scheduling should assign work in components more consistent with the schedule constraints perceived by line management. It should also reflect management's flexibility to vary crew sizes assigned to activities, as opposed to the mere scheduling of component activities, in efforts to level resource loads.

The development of improved aggregate scheduling techniques is deferred to a subsequent paper. In this paper, new techniques for individual project scheduling in a shipyard, motivated by the above discussion, are investigated.

## II. Construction Project Model

The model presented here was originally formulated in [2] and [4]. We present a simplified version of it. The construction project is modeled as a collection of component activities  $A$  utilizing a set  $K$  of exogenous resources. The dependencies among activities are represented by an activity-on-arc CPM network. Given are  $a_{ik}$ , the required amount of resource  $k \in K$  to complete activity  $i \in A$ , and  $d_i \geq 0$ , the minimum duration of activity  $i \in A$ . Let  $N$ , a subset of  $A$ , be the set of activities requiring non-zero resources, i.e.,  $a_{ik} > 0$  for some  $k \in K$ . Instantaneous application of resources is not possible, hence  $d_i > 0$  for  $i \in N$ . In the following sections, activities are assumed to belong to  $A$ , and resources to  $K$ , unless otherwise indicated.

Let  $d \equiv (d_1, \dots, d_{|A|})$  and  $t \equiv (t_1, \dots, t_{|A|})$ , where  $t$  is a vector of time assignment variables such that  $t \geq d$ . The rate of application of resource  $k$  to activity  $i \in N$ ,  $a_{ik}/t_i$ , is modeled as constant between the start and finish times of activity  $i$ . This represents a constant crew assignment to each activity  $i \in N$ . The crew size is determined by the activity time assignment  $t_i$ , the decision variable for each activity.

For a given time assignment vector  $t$  let  $CPM(t)$  denote the standard CPM scheduling computations of early (late) start and finish time and total activity slack. Consider  $CPM(d)$ . The computations determine the minimum project duration,  $T$ , and for each activity  $i$ , its slack,  $S_i$ . An activity  $i$  is *critical* if it is on a critical path, i.e.,  $S_i = 0$ . If, on the other hand,  $S_i > 0$  then activity  $i$  could operate for a longer period of time, using less resources per unit of time (if  $i \in N$ ), without delaying the project beyond  $T$ .

Let a *slack path* be defined as a maximal length chain of slack activities. In general, the total slack of a given slack path can be allocated among the activities on that path whereby all these activities become critical. A time assignment vector that eliminates all slack from the *slack subnetwork*, i.e., the subnetwork of slack activities, is said to be *critical*. Let  $\Lambda$ , be the set of critical time assignment vectors.

### III. Resource Leveling

Given time assignment vector  $r$  and  $CPM(r)$ , a load history can be computed for each resource. (This assumes rounding of non-integer time assignments  $r$ .) If  $r$  is critical then  $CPM(r)$  uniquely determines the resource load histories. If not, activity early (or late) starts, for example, could determine the schedule.

For  $k \in K$ , let  $c_k(r)$  be the required capacity of resource  $k$ , defined as the maximum load over all time periods  $t$ ,  $1 \leq t \leq T$ . Let  $p_k$  be the cost of maintaining a unit of capacity of resource  $k$  for the duration of the project.

We seek to minimize  $\sum_{k \in K} p_k c_k(r)$  over  $\Lambda$ , the set of all critical time assignment vectors  $r$ .

In [2] this problem was recognized as combinatorially too complex, and a heuristic procedure was proposed in an effort to obtain a near optimal solution. This paper presents an improvement of the procedure.

#### A. Peak Pricing Procedure

A sequence of problems is considered in which only the activities operating in time periods where resource loads are "close" to capacity  $c_k(r)$  are charged for their contribution to the capacity.

Let some time assignment vector  $r$  and the corresponding load histories be given. For each  $k \in K$ , define the peak intervals  $I_k$  as the set of time periods in which the load of resource  $k$  is within a specified percentage  $\beta$  of  $c_k(r)$ . Also, let  $I_{k,i}$  be the subset of  $I_k$  in which activity  $i \in N$  operates. Of course, activity  $i$  may operate partially or completely outside  $I_k$ . Note that  $I_{k,i} a_{k,i}(t)$  is the amount of resource  $k$  applied to activity  $i$  in  $I_k$ . Finally, let  $f_{k,i} \equiv |I_{k,i}|/|I_k|$  be the fraction of  $I_k$  in which activity  $i \in N$  is active. (The dependence on  $r$  has been suppressed in the notation  $I_k$ ,  $I_{k,i}$ ,  $f_{k,i}$ .) In [2], the fraction  $f_{k,i}$  was approximated by a 0/1 coefficient, that is,  $f_{k,i} = 1$  if and only if  $i$  operates in  $I_k$ .

The capacity  $c_k(r)$  can be approximated as follows:

$$\sum_{k \in K} \frac{f_{k,i} a_{k,i}(t)}{|I_k|} = \frac{\sum_{k \in K} I_{k,i} a_{k,i}(t)}{|I_k|}$$

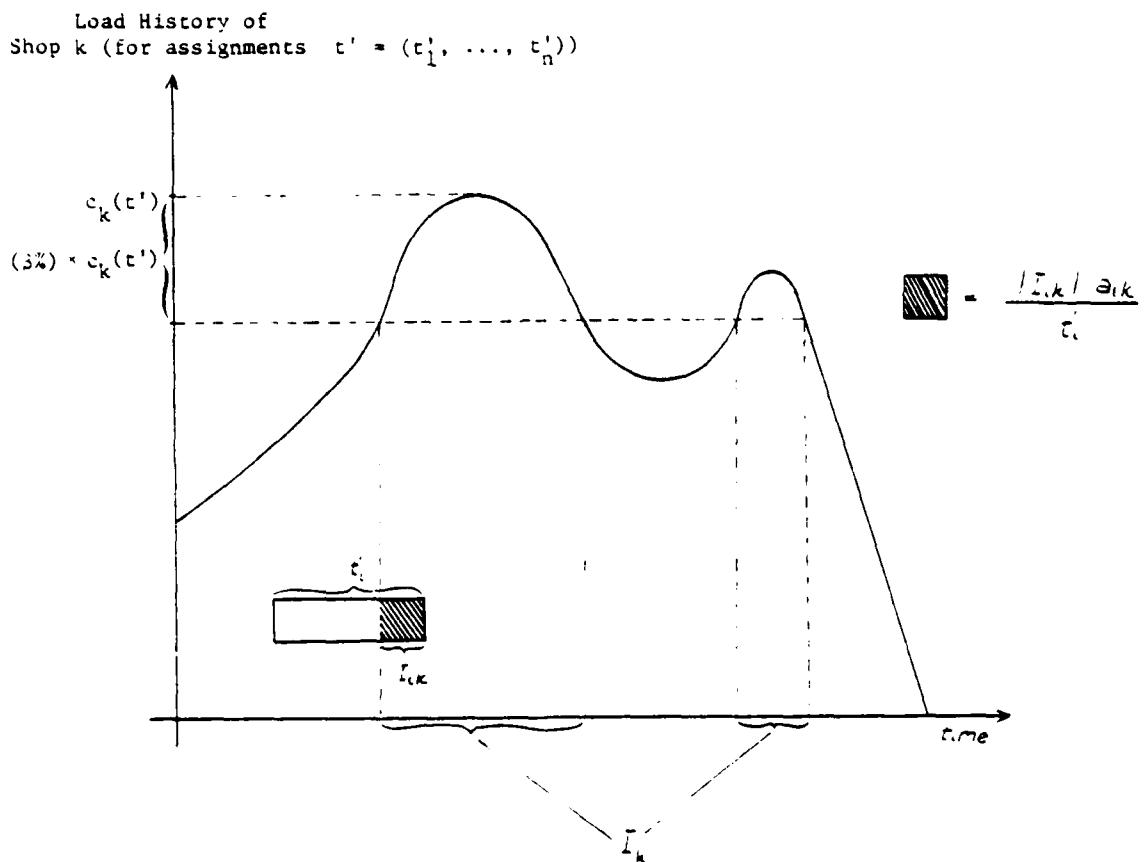
$\frac{\text{total amount of resource } k \text{ applied in } I_k}{\text{size of } I_k}$

$$\approx c_k(r).$$

Exhibit 3 depicts the above relationships graphically.

7

EXHIBIT 3  
RESOURCE PEAK INTERVALS



$c_k(t')$  = Peak load on Shop k for assignments  $t'$

$s$  = Peak interval tolerance

$I_k$  = Peak intervals, Shop k

$I_{ik}$  = Periods where activity  $i$  is in peak interval

$\blacksquare$  = Contribution of activity  $i$  to peak

If  $f_{ik}(t)$ , as a function of  $t$ , were readily computable we could perform

$$\min \left\{ \sum_{k \in K} p_k \sum_{i \in N} \frac{f_{ik}(t) a_{ik}}{t_i} : t \in \Lambda_i \right\}$$

to obtain an approximation of the minimum of  $\sum_{k \in K} p_k c_k(t)$ . This is not the case, so an iterative scheme is proposed.

Let some initial time assignment vector  $t'$  (for example  $d$ ) be given.

Step 1. Perform  $CPM(t')$  and compute the resource load histories.

Step 2. Compute  $c_k(t')$  for all  $k$ . If no improvement has occurred in  $\sum_{k \in K} p_k c_k(t')$ , stop.

Otherwise, determine  $l_k$  and  $f_{ik}$  for all  $i \in N, k \in K$ .

Step 3. Solve

$$P_i: \min \left\{ \sum_{k \in K} p_k \sum_{i \in N} \frac{f_{ik} a_{ik}}{l_i} : t \in \Lambda_i \right\}.$$

Let  $t^*$  be an optimal solution to  $P_i$ .

Set  $t \leftarrow t^*$  and go to Step 1.

## B. Optimization Problem Formulation

The mathematical program  $P_i$  can be formulated as a convex objective subject to linear constraints. Transformations of  $t$  allow  $\Lambda_i$ , the set of critical time assignment vectors, to be represented in two ways.

1. *Slack Allocation Variables:* Let  $s \equiv t-d$ ,  $s \equiv (s_1, \dots, s_4)$ . Then  $s_i$  is the slack allocated to activity  $i$  given the time assignment  $t \geq d$ .

Let  $\Lambda_s$  be the set of slack allocation vectors  $s$  satisfying

$$C_s: \sum_{i \in \pi} s_i = \text{slack of path } \pi \quad \begin{array}{l} \text{for all slack paths } \pi \\ \text{in the slack subnetwork} \end{array}$$

$$s_i = 0 \quad \text{for all critical activities } i$$

$$s \geq 0$$

Then  $t \in \Lambda_i$  if and only if  $s \in \Lambda_s$ . Problem  $P_i$  becomes

$$P_s: \min \left\{ \sum_{k \in K} p_k \sum_{i \in N} \frac{f_{ik} a_{ik}}{s_i + d} : s \in \Lambda_s \right\}.$$

Let  $s^*$  be an optimal solution to  $P_s$ . Then  $t^* = s^* + d$  is optimal for  $P_i$ . (This formulation was

presented in [2] with  $f_{ik}$  approximated as a 0/1 coefficient.)

The variables  $s_i$  for all critical activities  $i$  can be excluded from the problem. Furthermore, consider all slack paths which have only one element, i.e., an activity  $i$  by itself. If  $i$  does not occur in any other slack path then  $s_i$  can be set to its slack,  $S_i$ , thus eliminating a constraint and a variable. (See Exhibit 4a for an example of this.)

The number of linearly independent slack path constraints,  $n_{SP}$ , cannot exceed the number of slack allocation variables,  $n_{SA}$ , that remain, and may be considerably smaller (See Exhibit 4b).

The objective function coefficients in  $P_i$  (called *peak prices*) will be non-zero if and only if  $\sum_{k \in K} f_{ik} a_k > 0$ .

Let  $N(\tau')$  be the set of activities with non-zero peak prices, given time assignment vector  $\tau'$ . Slack allocation variables for activities not in  $N(\tau')$  can be eliminated by rewriting the constraint in  $C_i$  for slack path  $\pi$  as

$$\sum_{\substack{i \in \pi \\ i \in N(\tau')}} s_i \leq \text{slack of path } \pi.$$

$C_i$  is thereby reduced in size. However, the slack of path  $\pi$  that is not allocated in the optimal solution of  $P_i$ , if non-zero, must somehow be allocated to the excluded activities on  $\pi$  to obtain a complete CPM schedule and thus derive the load histories. (These allocation decisions were made implicitly, and arbitrarily, in the original formulation  $P_i$ .) Scheduling the excluded activities could be done, for example, at early (or late) starts with minimum time assignments. The resulting load histories will certainly be influenced by these slack allocation decisions.

2. *Event Time Variables*: Let  $V$  be the set of network nodes (events) and let  $(u(i), v(i))$  be the ordered pair of nodes representing arc (activity)  $i$ . Let  $\tau \equiv (\tau_1, \dots, \tau_{|V|})$  be an *event time vector*, where  $\tau_v$  is the point in time at which node  $v$  is reached. Let  $\Lambda$  be the set of event time vectors satisfying

$$C_{-i}: \tau_{v(i)} - \tau_{u(i)} \geq d, \quad \text{for all } i \in A$$

$$\tau_v = \tau'_v, \quad \text{for all nodes } v \in V \text{ on a critical path,}$$

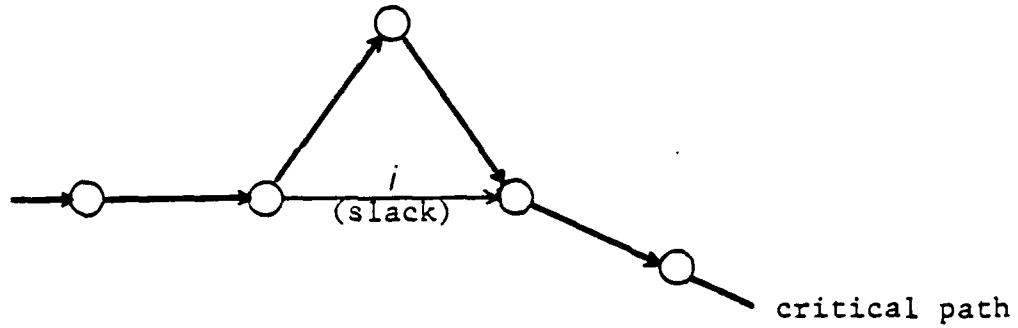
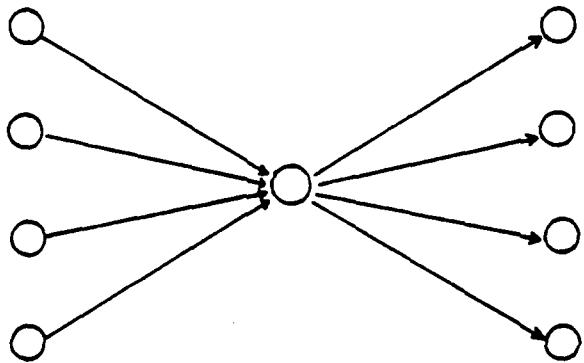
where  $\tau'_v$  is the critical node (event) time of  $v$ , as derived from  $CPM(d)$ .  $C_{-i}$  is similar to the standard CPM linear programming formulation. (See, for example, [3]). Then  $\tau \in \Lambda$ , if and only if  $\tau \in \Lambda_{-i}$ . Problem  $P_i$  becomes

$$P_i: \min \left\{ \sum_{k \in K} p_k \sum_{v \in V} \frac{f_{ik} a_k}{\tau_{v(i)} - \tau_{u(i)}} : \tau \in \Lambda_{-i} \right\}.$$

## EXHIBIT 4

(a) Single-Element Slack Paths

Activity  $i$  constitutes a "single-element slack path." Its slack can be pre-allocated.

(b) Redundant Slack Paths

This subnetwork of 6 slack activities gives rise to 16 slack path constraints (equalities), of which only 8 are linearly independent.

Let  $\tau^*$  be an optimal solution to  $P_r$ . Then  $\tau^* = \tau_{v(i)}^* - \tau_{u(i)}^*$  is optimal for  $P_r$ .

The event time variables  $\tau_v$  for critical nodes  $v$  can be excluded from the problem. Furthermore, constraints for all activities  $i$  such that  $u(i)$  and  $v(i)$  are critical nodes (as shown in Exhibit 4a) become redundant.

$C_r$  then consists of  $n_{SA}$  inequalities in  $n_{SV}$  variables, where  $n_{SV}$ , the number of slack nodes, is usually considerably smaller than  $n_{SP}$ . Also,  $n_{SV} < n_{SP}$ .

In its most compact form,  $C_r$  consists of  $n_{SP}$  inequalities in  $|N(t^*)|$  variables, plus non-negativity constraints. Exhibit 5 displays the constraints sets. The use of  $P_r$  or  $P_s$  will depend on the way in which the linear constraint set is imbedded in the non-linear programming algorithm. If the algorithm uses the dual constraint set in an imbedded linear program (see, for example, [1]) then  $P_r$  will most likely have the LP basis of smallest size.

### C. Procedure Termination and Cycling

The allocation of remaining slack for activities with zero peak prices is made arbitrarily in both formulations. This is a symptom of the inherent weakness of the Peak Pricing Procedure. Since  $j_{ik}$  does not change within an iteration as the time assignments (or their transformations) do, slack is allocated without feedback on the resulting shift in resource loads. A procedure can be imagined where feedback is more frequent, at the price of more frequent load history recomputations.

In general, no guarantee of improvement in each iteration of the procedure, let alone convergence to optimality, can be given. The new resource peaks will tend to be outside the previous peaks and may even be higher than before. In our experience a commonly occurring situation was the cycling between two different sets of peak loads. The following modification to the procedure was made to cope with this problem.

Given time assignment vectors  $t^1, t^2, \dots, t^L$  from previous iterations, define

$$f_k(t^1, \dots, t^L) \equiv \sum_{i=1}^L \alpha_{ki} f_k(t^i), \quad \text{where } \alpha_{ki} \geq 0, \quad \sum_{i=1}^L \alpha_{ki} = 1.$$

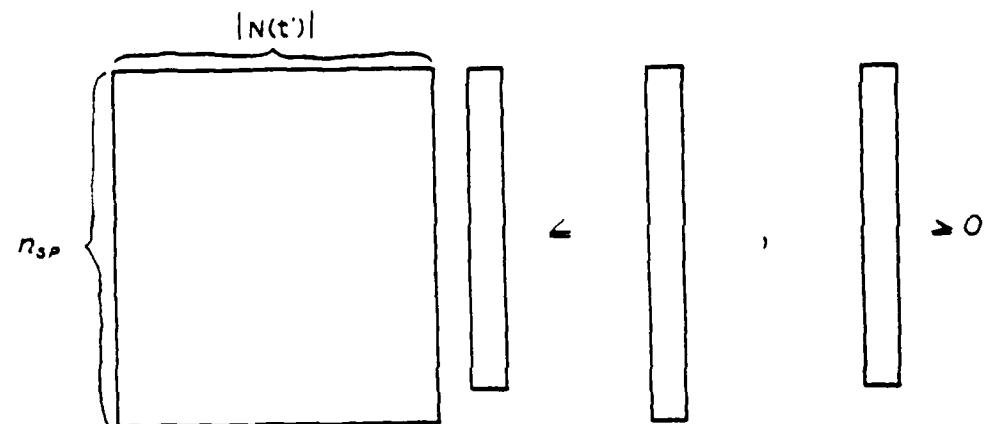
An obvious choice for  $\alpha_{ki}$  is

$$\alpha_{ki} = \frac{c_k(t^i)}{\sum_{i=1}^L c_k(t^i)},$$

which assigns weights to resource capacities according to their relative contribution over all  $L$  iterations.

EXHIBIT 5  
OPTIMIZATION PROBLEM CONSTRAINT SETS

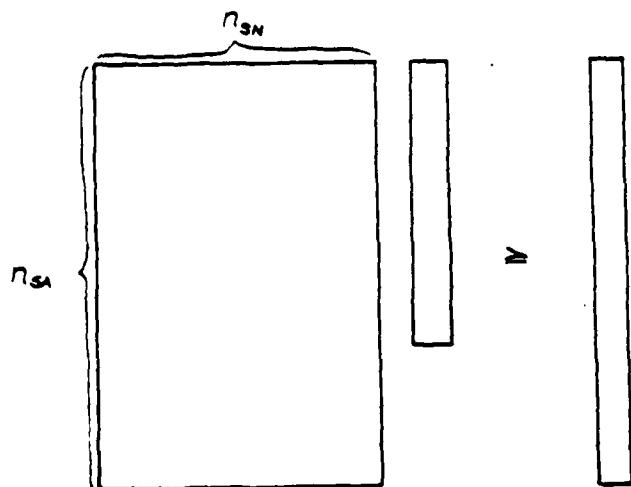
$C_S :$



$$n_{SN} \leq n_{SP} \leq n_{SA}$$

$$|N(t')| \leq n_{SA}$$

$C_T :$



The coefficients  $f_{ik}$  depend of course on the definition criterion for  $I_k$ . The criterion could vary, for example, from resource to resource, or from iteration to iteration, depending on the particular resource load pattern. Some experience with different criteria is described below.

#### D. Capacity Requirements vs. Project Delay

The formulations  $P_s$  and  $P_r$  can be adapted to study the trade-off between resource capacity cost and project delay. Let  $\gamma(T)$  be some project opportunity cost as a function of the variable project duration  $T$ . For simplicity of exposition, we assume  $x$  to be the only starting node and  $y$  the only finishing node in the network.

$P_s$  becomes

$$\begin{aligned} \min \sum_{k \in K} p_k \sum_{i \in V} \frac{f_{ik} a_{ik}}{s_i + d_i} + \gamma(T) \\ \text{s.t. } \sum_{i \in \pi} s_i = T - \sum_{i \in \pi} d_i \quad \text{for all paths } \pi \text{ from } x \text{ to } y \\ \text{in the network} \\ s \geq 0. \end{aligned}$$

and  $P_r$  becomes

$$\begin{aligned} \min \sum_{k \in K} p_k \sum_{i \in V} \frac{f_{ik} a_{ik}}{\tau_{v(i)} - \tau_{u(i)}} + \gamma(T) \\ \text{s.t. } \tau_{v(i)} - \tau_{u(i)} \geq d_i \quad \text{for all } i \in A \\ \tau_v - \tau_x = T. \end{aligned}$$

Since the durations of previously critical activities are no longer fixed, the variables and constraints associated with them can no longer be excluded a priori.

#### IV. Applications and Computational Experience

##### A. Network and Data

The original key-op network for a naval ship overhaul with about 1200 key-ops using 20 different labor types (resources) was selected for testing of the new techniques. Parallel or series key-ops that had similar work content (labor type) and dealt with the same technical subsystem, e.g., removal of various components of the main steam system, were grouped to form one aggregate activity. Attention was paid to combine key-ops with similar resource application rates. In this manner, a more aggregated network was developed, consisting of 316 activities using resources and 75 other activities (milestones, dummy activities, time lag activities). The largest twelve labor shops were retained with a total of 32,600 man-days of labor requirements. See Exhibit 6 for a list of shops and the labor requirements per shop. See Exhibit 7 for an example of an aggregate activity.

The minimum duration of each aggregate activity was computed as the maximum of the path lengths of the key-op activities which comprised the aggregate activity. The project duration was 211 days.

In the absence of good capacity cost estimates, the cost coefficients  $p_k$  were set to 1 and the objective became to minimize the sum of the shop capacity requirements.

The shops interact quite strongly. This is most evident in the case of shops 4 and 5. Shop 5 is responsible for removal, re-installation, and testing of mechanical subsystems. Shop 4 performs the on-shore repair of the subsystem components. The load pattern of shop 5 shows two regions of peak labor requirements: one during the removal phase, the second during re-installation. The peak of shop 4 lies in the valley between shop 5's peaks. (See Exhibit 8.)

The slack path formulation of the optimization step was chosen. To reduce the size of the problem those activities whose labor requirements per shop per day ( $a_{ik}/d$ ) did not exceed one man-day were excluded from the optimization problem  $P_k$ . It was reasoned that slack allocation to these activities would have reduced resource capacities only insignificantly. Ninety-eight activities with 3,600 man-days of labor requirements (11% of total requirements) were thus excluded. The constraint set  $C_k$  was reduced to 170 inequalities in 168 variables. An accelerated feasible conjugate direction algorithm due to [1] was used to solve the problems  $P_k$ .

##### B. Experiments

(a) Early experiments were performed with the 0/1 coefficients  $f_{ik}$  and various peak interval thresholds  $\beta$  ranging from 2% to 15%. Sequences of iterations with  $\beta$  varying from iteration to iteration led to inconclusive results. A value of  $\beta=5\%$  was then chosen for experiments

EXHIBIT 6  
LIST OF LABOR SHOPS

Numbers	Name	Project Labor Requirements (man-days)
1	Structural Group I	1330
2	Structural Group II	1390
3	Weld & Burn	2290
4	Mechanical Group-Shop	4310
5	Mechanical Group-Shipboard	5110
6	Boilermakers	2310
7	Electrical	3040
8	Pipe-fitting	5910
9	Electronics	2800
10	Shipwright	640
11	Painting	3000
12	Rigging	540
	Total Project Labor Requirements	<hr/> 32670

EXHIBIT 7  
EXAMPLE OF AGGREGATE ACTIVITY

Aggregate Activity: Inspect and Remove Fuel Oil System Components

Principal Shops Involved: Shop 8 - Pipefitting

Total Labor Requirements: 93 Man-days

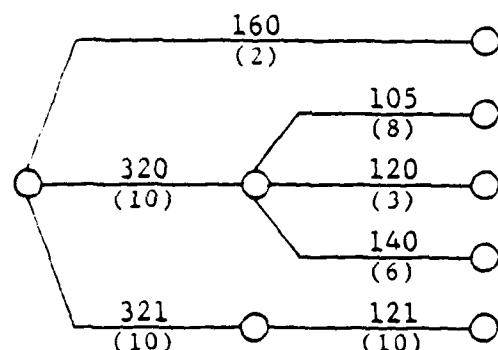
Minimum Duration: 20 days

Component Key-ops:

Key-op No.	Name	Minimum duration (days)	Total Labor requirements* (man-days)
105	Inspect Piping	8	10
120	Remove Relief Valves	3	2
121	Remove Strainers	10	30
140	Remove Pumps	6	8
160	Remove Controllers	2	2
320	Flush System	10	20
321	Flush Strainers	10	30
			<hr/>
			93

\*summed over all shops

Key-op Subnetwork



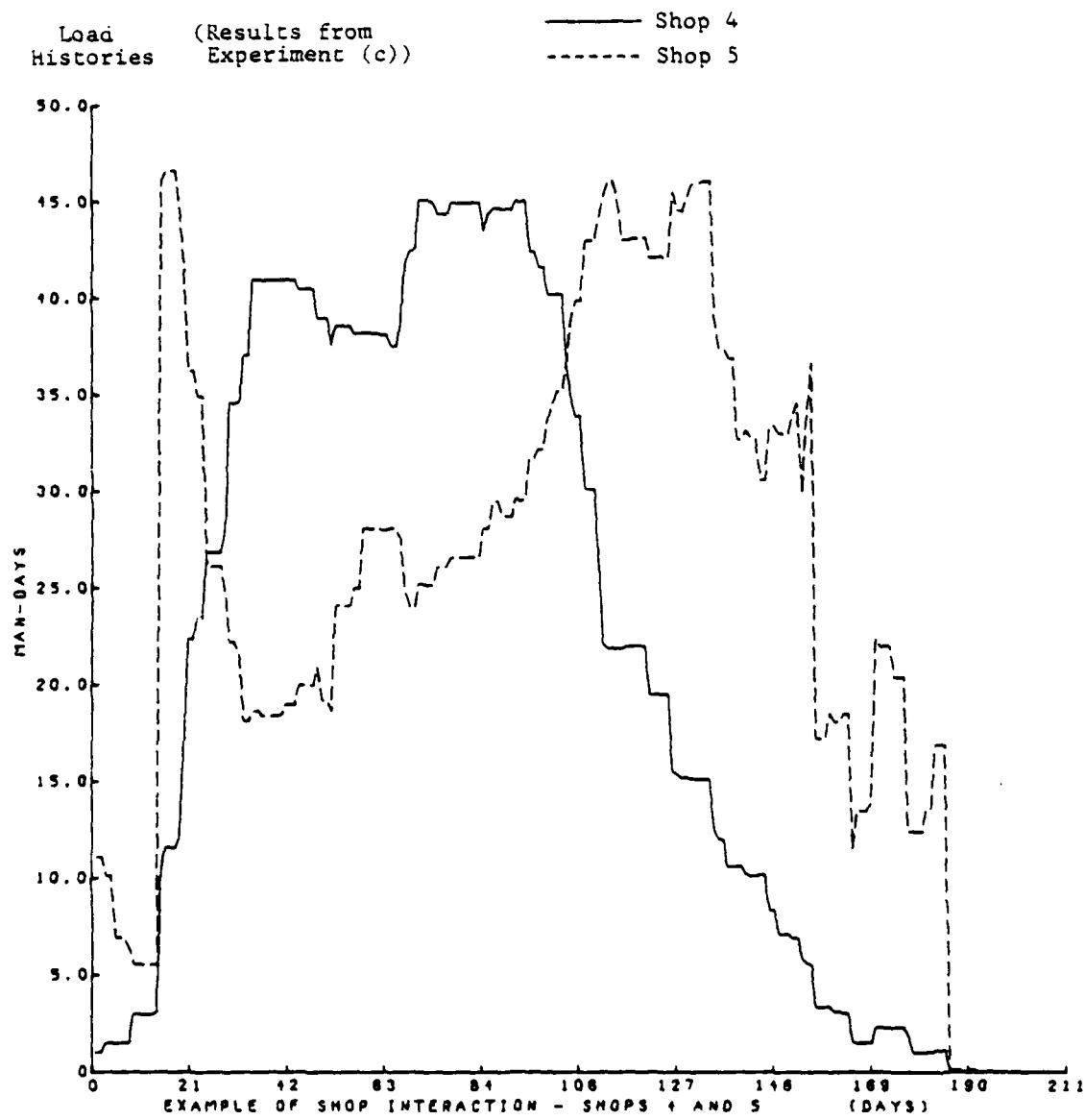
Legend:

(2) - minimum duration  
160 - key-op number

longest path

= minimum duration for aggregate activity

EXHIBIT 8  
EXAMPLE OF SHOP INTERACTION



(b) and (c).

(b) Initial load histories were computed from  $CPM(d)$  with an early start schedule. Then the slack of all single-element paths was allocated (as described in Section III.B.), which yielded an objective improvement of 12%. The improved definition of  $f_{ik}$  was used to compute peak prices for subsequent iterations of the Peak Pricing Procedure. Initially, four iterations were performed in succession. Pairs of successive iterations showed cycling behaviour. Multiple iteration weighting using iterations 1 and 2, and 3 and 4, was tried. The iteration with weighted peak prices from iterations 1 and 2 produced the best results thus far, an improvement of 31% over the initial solution. (See Exhibits 9 and 10.) Further iterations and weightings did not improve the solution.

(c) To test the effect of different initial load histories, a late start schedule from  $CPM(d)$  was used with load patterns markedly different from the early start schedule. As before, single-element path slack was pre-allocated. The first iteration produced a solution very close to the best result of (b). Individual shop capacities are similar in size (see Exhibit 9) and the load patterns have similar shape (See Exhibit 11 for the case of shop 8.)

(d) By setting  $\beta=100\%$ , the objective function of  $P_i$  becomes

$$\sum_{k \in K} p_k \sum_{e \in S_i} \frac{a_{ik}}{s_e + d_e},$$

so that  $P_i$  can now be stated as minimizing the sum of the resource application rates over all activities. The solution obtained was quite good compared with (b) and (c). (See Exhibit 9 for labor requirements by shop, and Exhibit 11 for a comparison of load patterns of shop 8.)

EXHIBIT 9  
EXPERIMENTAL RESULTS

Shop Capacity Requirements (Man-days)												Number of NLP Iter. *	Number of LP Iter. *	CPU Time (sec.) **	
1	2	3	4	5	6	7	8	9	10	11	12	Total			
<u>Experiment (b)</u>															
Early Start Schedule	23	17	35	66	67	45	33	60	26	14	55	9	450		
After Allocation of Slack of Single-Element Slack Paths	18	14	32	64	61	41	31	50	22	9	46	8	396		
After Iteration 1	13	11	24	47	64	21	25	47	18	9	41	7	327	14	251
After Iteration 2	13	11	25	49	53	38	24	47	18	9	42	7	334	7	143
After Weighting Peaks of Iterations 1 and 2	13	11	23	47	48	24	25	42	18	8	41	7	308	6	165
<u>Experiment (c)</u>															
Late Start Schedule	21	18	33	64	79	36	38	69	21	12	44	7	442		
After Allocation of Slack of Single-Element Slack Paths	17	16	29	62	73	37	32	57	21	9	55	6	394		
After Iteration 1	13	11	23	45	47	25	25	44	20	8	40	7	308	19	198
<u>Experiment (d)</u>															
Beta = 100%	13	12	22	47	54	27	25	43	19	8	40	7	317	13	453
															40.9

\* Possible Conjugate Direction Algorithm with Embedded LP [1]  
\*\* CDC 6400

EXHIBIT 10  
Experiment (b)  
SEQUENCE OF LOAD HISTORIES-SHOP 5

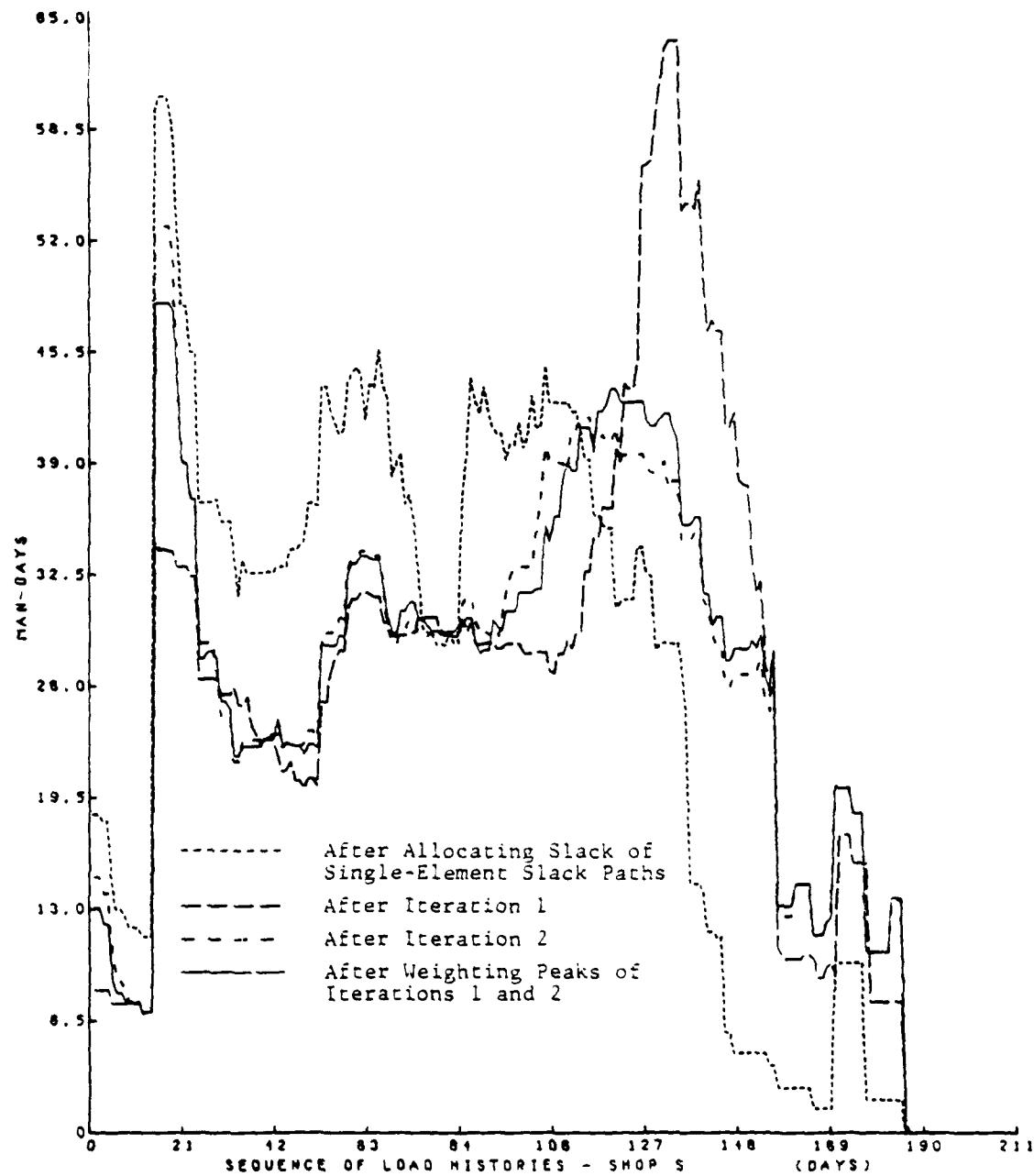
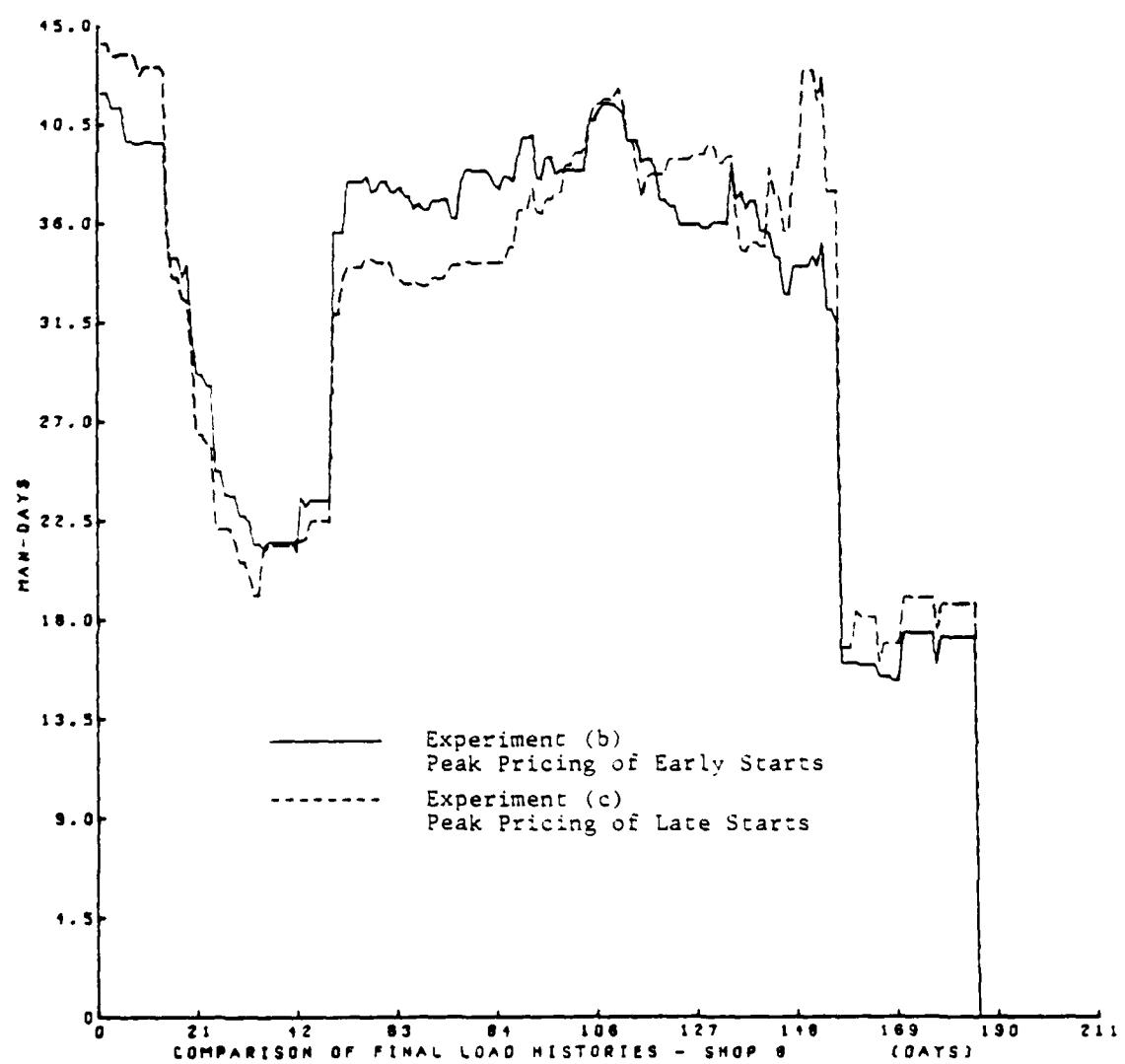


EXHIBIT 11  
COMPARISON OF SHOP 8 LOAD HISTORIES



## V. Summary and Analysis of Results

The development of new techniques for leveling resource loads of individual overhaul projects in a shipyard has been reviewed. The techniques exploit the crew size flexibility available to management in selecting time durations for aggregated activities.

Careful development of the overhaul data is mandatory. Activity labor requirements cannot be estimated solely from historical data, but depend on the specific work involved. Historical activity durations will certainly not represent minimum time assignments. Due to the assumption of constant activity loading, the minimum time assignments, together with the activity labor requirements, determine the maximum crew size assignable to each activity. Known crew size constraints may actually be the basis for the minimum time assignment estimates.

While optimal solutions cannot be generated, considerable loading improvement of initially selected schedules can be made. The effect of varying peak interval thresholds and multiple iteration weighting is not completely understood. Their effect will depend on the structure of the network. This is most likely the reason for the differences in performance between experiments (b) and (c).

Elimination of all slack from the network obviously increases the risk of project slippage. Additional constraints can be included to retain some amount of slack between milestones. In some cases, the final activity time assignments were unreasonably large. Time assignment upper bounds could remedy the problem of overstretched activities.

Varying activity durations (replanning) is an alternative to varying activity starting dates (rescheduling) as a means of resource leveling. However, replanning is not proposed as a complete substitute for rescheduling, in fact some rescheduling of the best replanning solutions obtained did improve resource loading, albeit slightly. Nonetheless, we submit that replanning of activities is a powerful and useful technique for leveling resource peaks in shipyards and related project industries.

## Glossary of Notation

$A$	- set of activities
$a_{ik}$	- labor requirement of activity $i$ , shop $k$
$\alpha_{kl}$	- peak price weighting factor
$\beta$	- peak interval threshold percentage
$C$	- constraint set of $P$
$c_k(t')$	- required capacity of shop $k$ given time assignment $t'$
$CPM(t')$	- CPM schedule computations given $t'$
$d_i$	- minimum time assignment of activity $i$ , $d \equiv (d_1, \dots, d_{14})$
$f_{ik}$	- fraction of $I_k$ in which activity $i$ operates
$I_k$	- subset of $I_k$ in which activity $i$ operates
$I_k'$	- set of time periods within $\beta$ of $c_k(t')$
$K$	- set of resources
$\Lambda$	- set of critical time assignment vectors
$N$	- set of activities with non-zero resource use
$N(t')$	- set of activities with non-zero peak prices, given $t'$
$n_{SA}$	- number of slack activities
$n_{SN}$	- number of slack nodes
$n_{SP}$	- number of slack paths
$P$	- optimization problem
$p_k$	- unit capacity cost for resource $k$
$S_i$	- slack of activity $i$ , given $CPM(d)$
$s_i$	- slack allocation to activity $i$ , $s \equiv (s_1, \dots, s_{14})$
$T$	- project duration
$t_i$	- time assignment of activity $i$ , $t \equiv (t_1, \dots, t_{14})$
$\tau_v$	- event time of node $v$ , $\tau \equiv (\tau_1, \dots, \tau_{14})$
$\tau_v'$	- event time of critical node $v$ , given $CPM(d)$
$(u(i), v(i))$	- node pair associated with activity $i$
$V$	- set of network nodes

### References

- [1] Best, Michael J. and Ritter, Klaus. "An Accelerated Conjugate Direction Method to Solve Linearly Constrained Minimization Problems." *Journal of Computer and Systems Sciences*. Vol. 11, No. 3 (1975).
- [2] Leachman, Robert C. "Aspects of Dynamic Production Planning." *Report ORC 79-8*. Operations Research Center, University of California, Berkeley (1979).
- [3] Moder, Joseph J. and Phillips, Cecil R., *Project Management with CPM and PERT*, Second Edition, Van Nostrand and Reinhold, New York (1970).
- [4] Shephard, Ronald W., Al-Ayat, Rokaya, and Leachman, Robert C. "Shipbuilding Production Function: An example of a Dynamic Production Function." *Report ORC 76-25*. Operations Research Center, University of California, Berkeley (1976).

**DAT**  
**FILM**